

Generar un esquema XSD que sirva para describir en XML dicha base de datos relacional.  
Asociar el documento XML correspondiente a dicha base de datos.

### Documento clase.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<clase xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
xsd:noNamespaceSchemaLocation="clase.xsd">
  <alumnos>
    <alumno cod="n12344345">
      <apenom>Alcalde García, Luis</apenom>
      <direc>Las Manos, 24</direc>
      <pobla>Lamadrid</pobla>
      <telef>942756645</telef>
    </alumno>
    <alumno cod="n43483437">
      <apenom>González Pérez, Olga</apenom>
      <direc>Miraflor 28 - 3A</direc>
      <pobla>Torres</pobla>
      <telef>942564355</telef>
    </alumno>
    <alumno cod="n88234942">
      <apenom>Fernández Díaz, María</apenom>
      <direc>Luisa Fernanda 53</direc>
      <pobla>Miera</pobla>
      <telef>942346945</telef>
    </alumno>
  </alumnos>
  <asignaturas>
    <asignatura cod="a1" nombre="FH"/>
    <asignatura cod="a2" nombre="FOL"/>
    <asignatura cod="a3" nombre="ISO"/>
    <asignatura cod="a4" nombre="LMSGI"/>
    <asignatura cod="a5" nombre="PAR"/>
    <asignatura cod="a6" nombre="GBD"/>
  </asignaturas>
  <notas>
    <nota alum="n12344345" asig="a1" calificacion="4"/>
    <nota alum="n43483437" asig="a1" calificacion="5"/>
    <nota alum="n88234942" asig="a1" calificacion="8"/>
    <nota alum="n12344345" asig="a2" calificacion="10"/>
    <nota alum="n43483437" asig="a2" calificacion="7"/>
    <nota alum="n88234942" asig="a2" calificacion="6"/>
    <nota alum="n12344345" asig="a3" calificacion="3"/>
    <nota alum="n88234942" asig="a3" calificacion="6"/>
    <nota alum="n12344345" asig="a4" calificacion="8"/>
    <nota alum="n43483437" asig="a4" calificacion="4"/>
    <nota alum="n12344345" asig="a5" calificacion="6"/>
    <nota alum="n12344345" asig="a6" calificacion="9"/>
  </notas>
</clase>
```

### Documento clase.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="Es-es">
      Base de datos de un instituto (Tarea 6 LMSGI)
    </documentation>
    <xsd:base>
      <xsd:element name="fecha" type="xsd:string"/>
    </base>
  </annotation>
</xsd:schema>
```

```

    <autor>José Luis Comesaña</autor>
  </xsd:documentation>
</xsd:annotation>
<xsd:element name="clase" type="claseDatos"/>
<xsd:complexType name="claseDatos">
  <xsd:sequence>
    <xsd:element name="alumnos" type="alumnosDatos" minOccurs="1"
maxOccurs="1"/>
    <xsd:element name="asignaturas" type="asignaturasDatos"
minOccurs="1" maxOccurs="1"/>
    <xsd:element name="notas" type="notasDatos" minOccurs="1"
maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="alumnosDatos">
  <xsd:sequence>
    <xsd:element name="alumno" type="alumnoDato" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="alumnoDato">
  <xsd:sequence>
    <xsd:element name="apenom" type="xsd:string"/>
    <xsd:element name="direc" type="xsd:string"/>
    <xsd:element name="pobla" type="xsd:string"/>
    <xsd:element name="telef" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:decimal">
          <xsd:minExclusive value="6000000000"/>
          <xsd:maxExclusive value="9999999999"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="cod" type="codAlumnos" use="required"/>
</xsd:complexType>
<xsd:simpleType name="codAlumnos">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[a-z]([0-9]){8}"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="asignaturasDatos">
  <xsd:sequence>
    <xsd:element name="asignatura" minOccurs="1"
maxOccurs="unbounded" type="asignaturaDato"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="asignaturaDato">
  <xsd:attribute name="cod" type="codAsignatura" use="required"/>
  <xsd:attribute name="nombre" type="nombreAsignatura"
use="required"/>
</xsd:complexType>
<xsd:simpleType name="codAsignatura">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[a-z][0-9]"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="nombreAsignatura">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[A-Z] +"/>
  </xsd:restriction>

```

```

</xsd:simpleType>
<xsd:complexType name="notasDatos">
  <xsd:sequence>
    <xsd:element name="nota" minOccurs="1" maxOccurs="unbounded"
type="notaDato"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="notaDato">

  <xsd:attribute name="alum" type="codAlumnos" use="required"/>
  <xsd:attribute name="asig" type="codAsignatura" use="required"/>
  <xsd:attribute name="calificacion" type="calAsignatura"
use="required"/>

  </xsd:complexType>
<xsd:simpleType name="calAsignatura">
  <xsd:restriction base="xsd:decimal">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="10"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Una vez realizado estos documentos realizar las siguientes consultas XQuery sobre los datos:

1. Obtener el nombre de todos los alumnos matriculados en algún módulo.

```

for $a in doc("clase.xml")//alumnos/alumno
where $a/@cod union $a/../../notas/nota/@alum
return
<alumno>{ data($a/apenom) }</alumno>

```

2. Obtener las calificaciones del alumno de código "n43483437" en cada módulo.

```

for $a in doc("clase.xml")//notas/nota
for $b in doc("clase.xml")//asignaturas/asignatura
where $a/@alum = "n43483437" and $b/@cod = $a/@asig
return
<alumno>{ data($b/@nombre), "---", data($a/@calificacion) }</alumno>

```

3. Obtener el nombre y el teléfono de cada alumno ordenado por apellidos de forma descendiente.

```

for $a in doc("clase.xml")//alumnos/alumno
let $b := index-of($a/apenom, ',')
order by $a/apenom descending
return
<alumno>{ data(substring-after($a/apenom, ',')), "---", data($a/telef)
}</alumno>

```

4. ¿Cuántos módulos hay?

```

let $a := doc("clase.xml")//asignaturas/asignatura
return
<alumno>{ "Existen: ", data(count($a)), "módulos distintos" }</alumno>

```

5. Obtener los nombres de los alumnos matriculados en LMSGI y sus notas, ordenado por notas.

```

for $a in doc("clase.xml")//asignaturas/asignatura,

```

```
$b in $a/../../notas/nota,
$c in $a/../../alumnos/alumno
where $a/@nombre="LMSGI" and $b/@asig = $a/@cod and $c/@cod = $b/@alum
order by $b/@calificacion
return
<alumno>{ "Nombre: ", data($c/apenom), " Calificacion: ",
data($b/@calificacion) }</alumno>
```

6. Obtener los nombres y las calificaciones de los matriculados en FH que han aprobado.

```
for $a in doc("clase.xml")//asignaturas/asignatura,
$b in $a/../../notas/nota,
$c in $a/../../alumnos/alumno
where $a/@nombre="FH" and $b/@asig = $a/@cod and $c/@cod = $b/@alum
and $b/@calificacion >=5
order by $b/@calificacion
return
<alumno>{ "Nombre: ", data($c/apenom), " Calificacion: ",
data($b/@calificacion) }</alumno>
```

Dado el fichero:

clase2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<fp>
  <ciclos_informatica>
    <ciclo id="ASIR">
      <nombre>Administracion de Sistemas Informaticos en Red</nombre>
      <legislacion>
        <real_decreto>Real Decreto 1629/2009, de 30 de
octubre</real_decreto>
        <orden>ORDEN de 19 de julio de 2010</orden>
      </legislacion>
      <grado>Superior</grado>
      <anyo>2009</anyo>
      <titulacion>Técnico Superior en Administración de Sistemas
Informáticos en Red</titulacion>
      <modulos>
        <modulo>
          <nombre>Implantación de sistemas operativos</nombre>
          <horas>224</horas>
        </modulo>
        <modulo>
          <nombre>Planificación y administración de redes</nombre>
          <horas>192</horas>
        </modulo>
        <modulo>
          <nombre>Fundamentos de hardware</nombre>
          <horas>96</horas>
        </modulo>
        <modulo>
          <nombre>Gestión de bases de datos</nombre>
          <horas>160</horas>
        </modulo>
        <modulo>
          <nombre>Lenguajes de marcas y sistemas de gestión de
información</nombre>
          <horas>96</horas>
        </modulo>
        <modulo>
          <nombre>Formación y orientación laboral</nombre>
```

```
<horas>96</horas>
</modulo>
</modulos>
</ciclo>
<ciclo id="DAM">
  <nombre>Desarrollo de Aplicaciones Multiplataforma</nombre>
  <legislacion>
    <real_decreto>Real Decreto 450/2010, de 16 de abril</real_decreto>
    <orden>ORDEN de 16 de junio de 2011</orden>
  </legislacion>
  <grado>Superior</grado>
  <anyo>2010</anyo>
  <titulacion>Técnico Superior en Desarrollo de Aplicaciones
Multiplataforma</titulacion>
  <modulos>
    <modulo>
      <nombre>Sistemas informáticos</nombre>
      <horas>160</horas>
    </modulo>
    <modulo>
      <nombre>Bases de datos</nombre>
      <horas>160</horas>
    </modulo>
    <modulo>
      <nombre>Programación</nombre>
      <horas>256</horas>
    </modulo>
    <modulo>
      <nombre>Lenguajes de marcas y sistemas de gestión de
información</nombre>
      <horas>96</horas>
    </modulo>
    <modulo>
      <nombre>Entornos de desarrollo</nombre>
      <horas>96</horas>
    </modulo>
    <modulo>
      <nombre>Formación y orientación Laboral</nombre>
      <horas></horas>
    </modulo>
  </modulos>
</ciclo>
<ciclo id="DAW">
  <nombre>Desarrollo de Aplicaciones Web</nombre>
  <legislacion>
    <real_decreto>Real Decreto 686/2010, de 20 de mayo</real_decreto>
    <orden>ORDEN de 16 de junio de 2011</orden>
  </legislacion>
  <grado>Superior</grado>
  <anyo>2010</anyo>
  <titulacion>Técnico Superior en Desarrollo de Aplicaciones
Web</titulacion>
  <modulos>
    <modulo>
      <nombre>Sistemas informáticos</nombre>
      <horas>160</horas>
    </modulo>
    <modulo>
      <nombre>Bases de datos</nombre>
      <horas>160</horas>
    </modulo>
```

```

<modulo>
  <nombre>Programación</nombre>
  <horas>256</horas>
</modulo>
<modulo>
  <nombre>Lenguajes de marcas y sistemas de gestión de
información</nombre>
  <horas>96</horas>
</modulo>
<modulo>
  <nombre>Entornos de desarrollo</nombre>
  <horas>96</horas>
</modulo>
<modulo>
  <nombre>Formación y orientación Laboral</nombre>
  <horas></horas>
</modulo>
</modulos>
</ciclo>
<ciclo id="SMR">
  <nombre>Sistemas Microinformaticos y Redes</nombre>
  <legislacion>
    <real_decreto>REAL      DECRETO      1691/2007,      de      14      de
diciembre</real_decreto>
    <orden>ORDEN de 7 de julio de 2009</orden>
  </legislacion>
  <grado>Medio</grado>
  <anyo>2007</anyo>
  <titulacion>Técnico      en      Sistemas      Microinformáticos      y
Redes</titulacion>
  <modulos>
    <modulo>
      <nombre>Montaje y mantenimiento de equipos</nombre>
      <horas>224</horas>
    </modulo>
    <modulo>
      <nombre>Redes locales</nombre>
      <horas>224</horas>
    </modulo>
    <modulo>
      <nombre>Aplicaciones ofimáticas</nombre>
      <horas>224</horas>
    </modulo>
    <modulo>
      <nombre>Sistemas operativos monopuesto</nombre>
      <horas>128</horas>
    </modulo>
    <modulo>
      <nombre>Formación y Orientación Laboral</nombre>
      <horas>96</horas>
    </modulo>
  </modulos>
</ciclo>
</ciclos_informatica>
</fp>

```

### Realizar los siguientes ejercicios:

1. Consulta que devuelve los nombres de los ciclos que tienen como identificador "DAM".

```

for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo
where $a/@id = "DAM"

```

```
return
<modulo>{ $a/modulos/modulo/nombre }</modulo>
```

- Consulta que devuelve las titulaciones de los ciclos que hayan sido publicadas a partir del año 2010 incluido, ordenados alfabéticamente por titulación.

```
for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo
where $a/any >= 2010
order by $a/titulacion
return
<modulo>{data($a/nombre)}{ $a/modulos/modulo/nombre }</modulo>
```

- Obtener el nombre de los módulos de todos los ciclos que tienen una duración de 96 horas.

```
for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo
order by $a
return
<ciclo>
<titulo>{data($a/nombre)}</titulo>
{for $b in $a/modulos/modulo
where number($b/horas) = 96
order by $b/nombre
return
<modulo>{data($b/nombre)}</modulo>}
</ciclo>
```

- Obtener las órdenes que regulan los ciclos de informática en Andalucía junto con el nombre de ciclo formativo.

```
for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo
order by $a
return
<ciclo>
<titulo>{data($a/nombre)}</titulo>
{for $b in $a/legislacion/orden
return
<orden>{data($b)}</orden>}
</ciclo>
```

- Consulta que obtiene para cada uno de los módulos el nombre y las horas.

```
for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo
order by $a
return
<ciclo>
<titulo>{data($a/nombre)}</titulo>
{for $b in $a/modulos/modulo
order by $b/nombre
return
<modulo>{data($b/nombre), " - ", data($b/horas), "horas"}</modulo>}
</ciclo>
```

- Devolver el nombre y las horas de los módulos ordenados desde el nombre del modulo más corto hasta el nombre del módulo más largo.

```
for
doc("clase2.xml")//fp/ciclos_informatica/ciclo/modulos/modulo
order by string-length($a/nombre)
return
<modulo>{data($a/nombre), " - ", data($a/horas), "horas"}</modulo>
```

7. Devolver una lista de titulaciones como atributo de una etiqueta ciclos.

```
for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo/titulacion
return
<ciclos tit="{data($a)}"></ciclos>
```

8. Devolver la suma de horas de los módulos del ciclo SMR excepto el modulo de "Formación y Orientación Laboral"

```
for $a in doc("clase2.xml")//fp/ciclos_informatica/ciclo[@id="SMR"]
return
let $b := $a/modulos/modulo[nombre != "Formación y Orientación Laboral"]
return
<hora>{sum($b/horas)}</hora>
```

### Anotaciones:

En algunos ejercicios he puesto más de lo solicitado únicamente por temas de presentación y práctica, no por necesidad para el correcto funcionamiento de dicho ejercicio.